138 13/11/23

13 NOV-23

Scoring Indicators

COURSE NAME : Fundamentals of Artificial Intelligence and Machine Learning

COURSE CODE : TED (21) – 5133C

QID : 2109230301A

A

PART A

I. Answer all the following questions in one word or sentence.

(9 x 1 = 9 Marks)

Max. marks

Q.No.	Scoring Indicators	Split score	Sub Total	Total score
	PART A			9
I.1	The science and engineering of **making intelligent machines**, especially intelligent computer programs		1	
I.2	Gaming, Natural Language Processing, Expert Systems, Vision Systems, Speech Recognition, Intelligent Robots etc(any two)		1	
I.3	The dictionary is an unordered collection that contains key:value pairs separated by commas inside curly brackets.		1	
I.4	if condition: # execute code block		1	
I.5	Regression analysis is a statistical method to model the relationship between a dependent (target) and independent (predictor) variables with one or more independent variables		1	
I.6	Supervised machine learning algorithms and Unsupervised machine learning algorithms		1	
I.7	k' in KNN is a parameter that refers to the **number of nearest neighbours**		1	
I.8	Uninformed (Blind search) search and Informed search (Heuristic search) algorithms.		1	
I.9	Mini-max algorithm		1	
	PART B			24
II.1	Gaming, Natural Language Processing,	1+1+1	3	

	Expert Systems, Vision Systems, Speech Recognition, Intelligent Robots Handwriting Recognition (list and define any 3)			
II.2	1 Auditory Learning 2.Episodic Learning 3. Motor Learning, 4.Observational Learning 5.Perceptual Learning 6. Relational Learning	0.5x6	3	
II.3	A for loop can be used in the following two situations: - Repeat a set of statements a finite number of times. - Iterate through a string, list, tuple, set or dictionary. To repeat a set of statements a finite number of times a built-in function range( ) is used. range( ) function generates a sequence of integers. eg; for i in range(1, 10, 2) : print(i, i * i, i * i * i)	2+1	3	
II.4	Number = int(input("Please Enter any Number: ")) Reverse = 0 while(Number > 0): Reminder = Number %10 Reverse = (Reverse *10) + Reminder Number = Number //10 print("\n Reverse of entered number is = %d" %Reverse)	3	3	
II.5	Simple syntax & less coding Testing can be easier Inbuilt libraries for AI projects. Open source Python can be used for a broad range of programming tasks like small shell script to enterprise web applications.	3	3	

II.6	# create a list of 5 odd numbers
	a = [1, 3, 5, 7, 9]
		1+1+1	3	

	print(a)
	# create a list of 5 even numbers
	b = [2, 4, 6, 8, 10]
	print(b)
	# combine the two lists
	a = a + b
	print(a)			

II.7	Step 1: Importing ML Library for Python Step 2: Importing the Dataset Step 3: Organizing the Data into Training and Testing Sets Step 4: Build the model Step 5: Train the Model Step 6: Making Predictions on Test Set. Step 7: Evaluating the Classifier Accuracy	3		
II.8	# importing libraries pandas scipy numpy from sklearn.preprocessing import MinMaxScaler seaborn matplotlib.pyplot (not necessary)	3		
II.9	A search problem consists of: A State Space. Set of all possible states where you can be. A Start State. The state from where the search begins. A Goal State. A function that looks at the current state returns whether or not it is the goal state. The Solution to a search problem is a sequence of actions, called the plan that transforms the start state to the goal state.	3		

II.10	from easyAI import TwoPlayersGame, AI_Player, Negamax from easyAI.Player import Human_Player			
	PART C			4 2
III	Machine Learning, Searching, Artificial neural networks, Genetic Algorithm, logic, Knowledge Representation Deep Learning	1x7	7	
IV	AI can learn through data. AI can teach itself . AI can respond in real time AI achieves accuracy AI can organize data to get most out of it	5+2	7	
V	Basic types - int, float, complex, bool,string, bytes	7	7	

	# int can be expressed in binary, decimal, octal, hexadecimal # binary starts with 0b/0B, octal with 0o/0O, hex with 0x/0X 0b10111, 156, 0o432, 0x4A3 # float can be expressed in fractional or exponential form - 314.1528, 3.141528e2, 3.141528E2 # complex contains real and imaginary part 3 + 2j, 1 + 4J # bool can take any of the two Boolean values both starting in caps True, False # string is an immutable collection of Unicode characters enclosed # within ' ', " " or """ """. 'Name', "Name", """Name""" # bytes represent binary data b'\xa1\xe4\x56' # represents 3 bytes with hex values a1a456			

VI	import operator
	d = {'Oil' : 230, 'Clip' : 150, 'Stud' : 175, 'Nut' : 35}
	print('Original dictionary : ', d)
	# sorting by key
	d1 = sorted(d.items( ))
	print('Asc. order by key : ', d1)
	d2 = sorted(d.items( ), reverse = True)
	print('Des. order by key : ', d2)
	# sorting by value
	d1 = sorted(d.items( ), key = operator.itemgetter(1))
	print('Asc. order by value : ', d1)
	d2 = sorted(d.items( ), key = operator.itemgetter(1),
		7	7	

	reverse = True) print('Des. order by value : ', d2)			
VII	Lists are used to store multiple items in a single variable.Lists are one of 4 built-in data types in Python used to store collections of data. Lists are created using square brackets: thislist = ["apple", "banana", "cherry"] print(thislist) Accessing elements from the List In order to access the list items refer to the index number. Use the index operator [ ] to access an item in a list. The index must be an integer. Nested lists are accessed using nested indexing. ftlist = ["apple", "banana", "cherry"] print(ftlist[0])	3.5+3.5	7	

VIII	string = input("Enter your string :")
	uppercase_string = string.upper()
	print("Uppercase string:", uppercase_string)
	
	# to count the occurrence of a character in a string
	string = input ("Enter a sentence :")
	character = input("Enter the Character: ")
	count = string.count(character)
	print("Count:", count)
	
	# to split a string into a list of words
	string = input ("Enter a sentence :")
	words = string.split()
	print("List of words:", words)
		2+ 2.5 + 2.5	7	

IX	Linear regression analysis is used to predict the value of a variable based on the value of another variable. The variable you want to predict is called the dependent variable. The variable you are using to predict the other variable's value is called the independent variable. Market analysis. Financial analysis. Sports analysis. Environmental health. Medicine. Predicting outcomes.	2+5	7	
X	Binarization This is the preprocessing technique which is used when we need to convert our numerical values into Boolean values. We can use an inbuilt method to binarize the input data say by using 0.5 as the threshold value in the following way: data_binarized = preprocessing.Binarizer(threshold=0.5).transform(input_data) print("\nBinarized data:\n", data_binarized) Now, after running the above code we will get the following output, all the values above 0.5(threshold value) would be converted to 1 and all the values below 0.5 would be converted to 0. Binarized data [[1. 0. 1.] [0. 1. 1.] [0. 0. 1.] [1. 1. 0.]] Mean Removal It is another very common preprocessing technique that is	7		

used in machine learning.

Basically it is used to eliminate the mean from feature vector so that every feature is

centered on zero. We can also remove the bias from the features in the feature vector.

For applying mean removal preprocessing technique on the sample data, we can write the

Python code shown below. The code will display the Mean and Standard deviation of the

input data:

```
print("Mean =", input_data.mean(axis=0))
print("Std deviation = ", input_data.std(axis=0))
```

We will get the following output after running the above lines of code:

```
Mean = [ 1.75 -1.275 2.2 ]
Std deviation = [ 2.71431391 4.20022321 4.69414529]
```

Scaling

It is another data preprocessing technique that is used to scale the feature vectors. Scaling

of feature vectors is needed because the values of every feature can vary between many

random values. In other words we can say that scaling is important because we do not

want any feature to be synthetically large or small. With the help of the following Python

code, we can do the scaling of our input data, i.e., feature vector:

```
# Min max scaling
data_scaler_minmax =
preprocessing.MinMaxScaler(feature_range=(0,1))
data_scaled_minmax =
data_scaler_minmax.fit_transform(input_data)
print ("\nMin max scaled data:\n", data_scaled_minmax)
```

We will get the following output after running the above lines of code:

Min max scaled data

[[0.48648649 0.58252427 0.99122807]

[0. 1. 0.81578947]

[0.27027027 0. 1.]

[1. 0.99029126 0.]]

Normalization

It is another data preprocessing technique that is used to modify the feature vectors. Such

kind of modification is necessary to measure the feature vectors on a common scale.

Followings are two types of normalization which can be used in machine learning:

L1 Normalization

It is also referred to as Least Absolute Deviations. This kind of normalization modifies

the values so that the sum of the absolute values is always up to 1 in each row. It can be

implemented on the input data with the help of the following Python code:

```
# Normalize data
data_normalized_l1 = preprocessing.normalize(input_data, norm='l1')
```

L2 Normalization

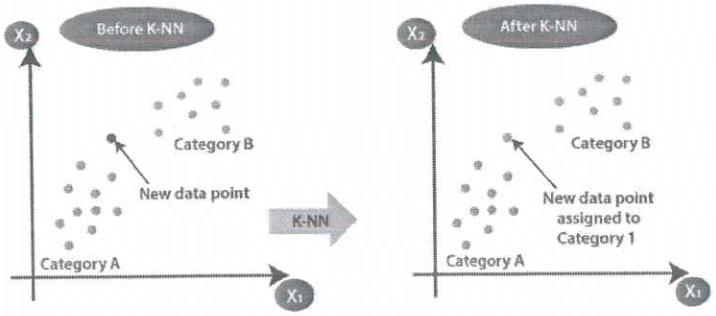
It is also referred to as least squares. This kind of normalization modifies the values so

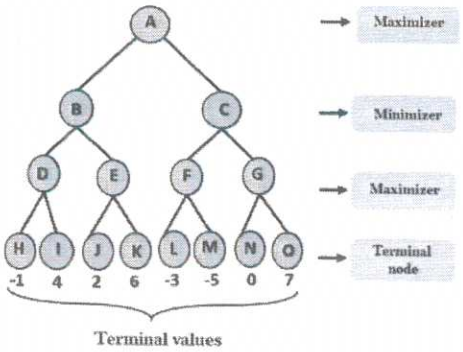
that the sum of the squares is always up to 1 in each row. It can be implemented on the

input data with the help of the following Python code:

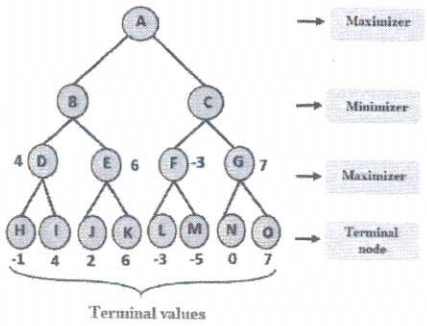
```
# Normalize data
data_normalized_l2 = preprocessing.normalize(input_data, norm='l2')
print("\nL2 normalized data:\n", data_normalized_l2)
```

XI		Classification	Regressor	1x7	7
	1.	In this problem statement, the target variables are discrete.	In this problem statement, the target variables are continuous.		
	2.	Problems like Spam Email Classification, Disease prediction like problems are solved using Classification Algorithms.	Problems like House Price Prediction, Rainfall Prediction like problems are solved using regression Algorithms.		
	3.	In this algorithm, we try to find the best possible decision boundary which can separate the two classes with the maximum possible separation.	In this algorithm, we try to find the best-fit line which can represent the overall trend in the data.		
	4.	Evaluation metrics like Precision, Recall, and F1-Score are used here to evaluate the performance of the classification algorithms.	Evaluation metrics like Mean Squared Error, R2-Score, and MAPE are used here to evaluate the performance of the regression algorithms.		
	5.	Here we face the problems like binary Classification or Multi-Class Classification problems.	Here we face the problems like Linear Regression models as well as non-linear models.		
	6.	Input Data are independent variables and categorical dependent variable.	Input Data are independent variables and continuous dependent variable.		
	7.	Output is Categorical labels.	Output is Continuous numerical values.		
XII	K-Nearest Neighbour is one of the simplest Machine Learning algorithms based on Supervised Learning technique. K-NN algorithm assumes the similarity between the new case/data and available cases and put the new case into the category that is most similar to the available categories. K-NN algorithm stores all the available data and classifies a new data point based on the similarity. This means when new data appears then it can be easily classified into a well suite category by using K- NN algorithm. K-NN algorithm can be used for Regression as well as for Classification but mostly it is used for the Classification problems. K-NN is a non-parametric algorithm, which means it does not make any assumption on underlying data. It is also called a lazy learner algorithm because it does not learn from the training set immediately instead it stores the dataset and at the time of classification, it performs an action on the dataset. KNN algorithm at the training phase just stores the dataset and when it gets new data, then it classifies that data into a category that is much similar to the new data.		7		

				
XIII	There are two players one is called Maximizer and other is called Minimizer. Maximizer will try to get the Maximum possible score, and Minimizer will try to get the minimum possible score. This algorithm applies DFS, so in this game-tree, we have to go all the way through the leaves to reach the terminal nodes. At the terminal node, the terminal values are given so we will compare those value and backtrack the tree until the initial state occurs. Following are the main steps involved in solving the two-player game tree: Step-1: In the first step, the algorithm generates the entire game-tree and apply the utility function to get the utility values for the terminal states. In the below tree diagram, let's take A is the initial state of the tree. Suppose maximizer takes first turn which has worst-case initial value =- infinity, and minimizer will take next turn which has worst-case initial value = +infinity.	4+3	7	



Step 2: Now, first we find the utilities value for the Maximizer, its initial value is -∞, so we will compare each value in terminal state with initial value of Maximizer and determines the higher nodes values. It will find the maximum among the all.



For node D max(-1,- -∞) => max(-1,4)= 4

For Node E max(2, -∞) => max(2, 6)= 6

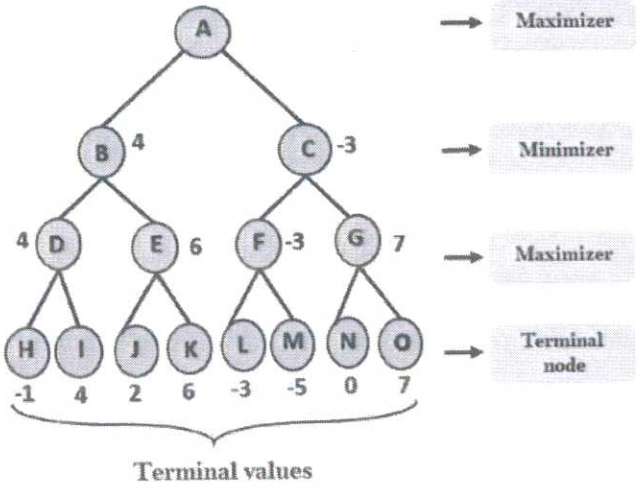
For Node F max(-3, -∞) => max(-3,-5) = -3

For node G max(0, -∞) = max(0, 7) = 7

Step 3: In the next step, it's a turn for minimizer, so it will compare all nodes value with +∞, and will find the 3rd layer node values.

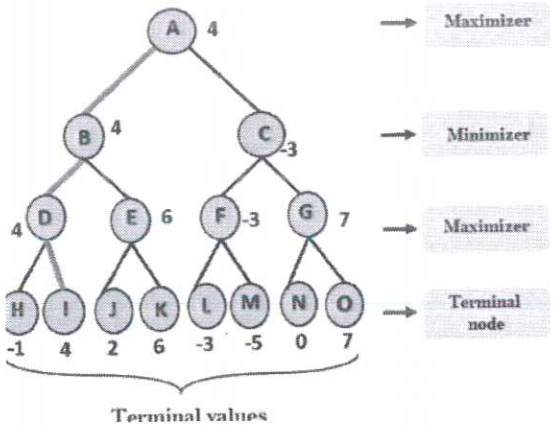
For node B= min(4,6) = 4

For node C= min (-3, 7) = -3



Step 4: Now it's a turn for Maximizer, and it will again choose the maximum of all nodes value and find the maximum value for the root node. In this game tree, there are only 4 layers, hence we reach immediately to the root node, but in real games, there will be more than 4 layers.

For node A max(4, -3)= 4



XIV	In this game, there would be a pile of coins. Each player has to take a number of coins from that pile. The goal of the game is to avoid taking the last coin in the pile. We will be using the class LastCoinStanding inherited from the TwoPlayersGame class of the easyAI library. • Import the required packages • Inherit the class from the TwoPlayerGame class to handle all operations of the game • Define the players and the player who is going to start the game. • Define the number of coins in the game	7	7	

	• Define the maximum number of coins a player can take in a movie. • Define possible moves. • Define the removal of the coins • Define the removal of the coins • Define when to stop the game, that is when somebody wins. • Define how to compute the score. • Define the number of coins remaining in the pile.			